



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2003-0080117
Application Number

출원 년 월 일 : 2003년 11월 13일
Date of Application NOV 13, 2003

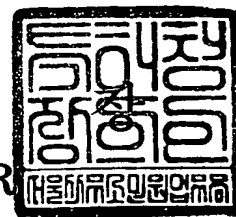
출원인 : 한국전자통신연구원
Applicant(s) Electronics and Telecommunications Research Inst



2003 년 11 월 24 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0003
【제출일자】	2003.11.13
【발명의 명칭】	이동 단말 장치에서의 움직임 검출 방법
【발명의 영문명칭】	METHOD FOR DETECTING MOTION IN MOBILE DEVICE
【출원인】	
【명칭】	한국전자통신연구원
【출원인코드】	3-1998-007763-8
【대리인】	
【성명】	권태복
【대리인코드】	9-2001-000347-1
【포괄위임등록번호】	2001-057650-1
【대리인】	
【성명】	이화익
【대리인코드】	9-1998-000417-9
【포괄위임등록번호】	1999-021997-1
【발명자】	
【성명의 국문표기】	김선태
【성명의 영문표기】	KIM, SEON-TAE
【주민등록번호】	701005-1637321
【우편번호】	305-752
【주소】	대전광역시 유성구 송강동 8-2 청솔아파트 103동 1505호
【국적】	KR
【발명자】	
【성명의 국문표기】	김범호
【성명의 영문표기】	KIM, Bumho
【주민등록번호】	740621-1074519
【우편번호】	137-063
【주소】	서울특별시 서초구 방배3동 541-3 명빌라 2동 102호
【국적】	KR

【발명자】**【성명의 국문표기】** 마평수**【성명의 영문표기】** MAH, Pyeong Soo**【주민등록번호】** 610829-1481310**【우편번호】** 305-340**【주소】** 대전광역시 유성구 도룡동 391 타운하우스 3동 201호**【국적】** KR**【심사청구】** 청구

【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인
권태복 (인) 대리인
이화익 (인)

【수수료】**【기본출원료】** 20 면 29,000 원**【가산출원료】** 2 면 2,000 원**【우선권주장료】** 0 건 0 원**【심사청구료】** 6 항 301,000 원**【합계】** 332,000 원**【감면사유】** 정부출연연구기관**【감면후 수수료】** 166,000 원**【기술이전】****【기술양도】** 희망**【실시권 허여】** 희망**【기술지도】** 희망**【첨부서류】** 1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 적은 메모리 공간을 가진 이동 단말 장치와 같은 임베디드 시스템에서 많은 메모리를 필요로 하는 비디오 디지털 신호처리 알고리즘이 동작될 수 있도록 움직임 검출을 효율적으로 구현하는 방법에 관한 것이다. 본 발명에서는 현재의 매크로 블록과 다음의 매크로 블록간의 참조 탐색 영역의 중복성이 존재함을 인지하여, 필요 메모리의 데이터 이동을 최소화시켜 내부 메모리를 효율적으로 이용하기 위한 것이다. 이와 같이, 움직임 검출에서 적은 내부 메모리를 이용함으로써 내부 메모리를 다른 용도로 사용할 수 있는 가용성을 증대시키고, 내부 메모리에서 보다 많은 연산을 수행함으로써 빠른 성능으로 복잡한 MPEG-4 비디오 부호화 알고리즘이 구현될 수 있다.

【대표도】

도 3

【색인어】

움직임 검출, 전력소모, 내부 메모리, 이동 단말 장치

【명세서】**【발명의 명칭】**

이동 단말 장치에서의 움직임 검출 방법{METHOD FOR DETECTING MOTION IN MOBILE DEVICE}

【도면의 간단한 설명】

도 1은 일반적인 MPEG-4 비디오 부호기의 개략적인 구성을 나타낸 도면.

도 2a 및 도 2b는 MPEG-4 비디오 부호기 알고리즘에 따른 기존의 움직임 검출 방식을 설명하는 도면.

도 3은 본 발명의 실시예에 따른 이동 단말 장치에서의 움직임 검출 방법이 적용되는 메모리 사용예를 나타낸 도면.

도 4a 및 도 4b는 본 발명의 실시예에 따른 움직임 검출 방법이 적용되는 메모리 사용예와 기존의 방법에 따른 메모리 사용예를 비교하여 나타낸 도면.

도 5는 본 발명의 실시예에 따른 움직임 검출 방법에서 프레임의 매크로 블록의 움직임이 세로 방향으로 진행하는 경우와 가로 방향으로 진행하는 경우의 버퍼의 할당과 데이터의 이동을 각각 나타낸 도면.

도 6은 상기 도 5에서 매크로 블록의 움직임이 세로 방향으로 진행할 경우의 본 발명의 실시예에 따른 움직임 검출 방법의 처리 과정을 나타낸 순서도.

도 7a 및 도 7b는 본 발명의 실시예에 따른 움직임 검출 방법에서 가로 방향과 세로 방향의 움직임 검출시 사용되는 버퍼의 데이터 배열 상태를 나타낸 도면.

도 8a 및 도 8b는 하나의 매크로 블록의 움직임 SAD 값을 구하는 과정을 나타낸 도면.

도 9는 참조 탐색 영역이 $[-8, 7]$ 인 상태에서 본 발명의 실시예에 따른 움직임 검출 방법이 실행될 때의 참조 탐색 영역을 위한 버퍼의 할당과 매크로 블록 간의 관계를 나타낸 도면.

도 10a 및 도 10b는 참조 탐색 구간이 $[-16, 15]$ 인 탐색 구간에서 사용한 순환 버퍼가 참조 탐색 구간 $[-8, 7]$ 에 그대로 사용될 때, 움직임 검출시의 탐색 영역과 찾고자 하는 움직임 검출 매크로 블록 간의 관계를 나타낸 도면.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<11> 본 발명은 이동 단말 장치의 비디오 압축 알고리즘에서 사용되는 움직임 검출 방법에 관한 것으로서, 더욱 상세하게는 이동 단말 장치와 같이 적은 용량의 메모리를 가진 장치에서 MPEG-4 비디오 부호기를 구현함에 있어서 비디오 압축의 많은 부분을 차지하는 움직임 검출 알고리즘을 효과적으로 구현하기 위한 방법에 관한 것이다.

<12> 임베디드 시스템에서 현재 사용되고 있는 저 비트율 비디오 압축 알고리즘은 표준 MPEG-4로서, 그 기본 구조는 도 1에 도시된 바와 같이, 일반 비디오에서 사용되는 사각형의 텍스처 부호화부(1)와 임의의 형상을 압축하는 형상 부호화부(2)로 구성되어 있으며, 다양한 응용에 대처할 수 있도록 되어있다. 이 중에서 텍스처 부호화부(1)는 공간 정보와 시간 정보에 대한 불필요한 데이터를 압축하기 위한 것으로, 현재 다양한 표준의 공통 알고리즘으로 사용된다.

<13> 이 알고리즘은 공간과 시간에 대한 정보를 압축하는 것으로, 공간 정보를 압축하기 위해서는 이산 역현 변환 및 역변환(DCT/IDCT) 과정, 그리고 양자화 과정이 요구된다. 또한, 시간

공간에 대한 압축을 수행하기 위해서는 움직임 검출과 움직임 보상, 그리고 이전 화면의 저장을 위한 버퍼링 과정이 필요하다. 가변길이 부호화부는 공간 정보와 시간 정보에 대해서 양자화된 값을 정보의 확률에 따라 압축하는 동작을 수행한다.

<14> MPEG-4의 움직임 검출부(10)는 상당히 많은 메모리 접근(access) 및 메모리 크기를 필요로 하여 MPEG-4 비디오 부호기 알고리즘의 성능에 지대한 영향을 미친다. 움직임 검출부(10)는 프레임의 방향에 따라 P, B프레임 형식으로 나뉘지만, 메모리 접근에 의한 움직임 검출방식은 비슷하므로 본 발명에서는 P프레임에 대해서만 언급한다. P프레임에 대한 움직임 검출부(10)의 동작은 도 2a에 도시된 바와 같이, 이전 프레임(t-1)과 현재 프레임(t)으로 수행된다. 즉, 상기 움직임 검출부(10)는 현재 프레임의 매크로 블록에 대해서 이전 프레임의 탐색 영역 내에서 픽셀단위로 움직여 가장 유사한 위치를 찾아낸다. 유사도를 찾는 방법은 평균 제곱 오차법(MSE : Mean Square Error, 이하 'MSE'라 함), 절대값차 합산법(SAD : Sum of Absolute Difference, 이하 'SAD'라 함) 또는 평균 절대값 차이법(MAD : Mean Absolute Difference, 이하 'MAD'라 함) 등의 방법이 있다.

<15> 도 2a 및 도 2b는 MPEG-4 비디오 부호기 알고리즘에 따른 기존의 움직임 검출 방식을 설명하는 도면으로, 도 2a에 도시된 방식은 참조 프레임과 현재의 프레임을 외부 메모리에 저장하여 유사도를 찾으며, 비디오의 프레임 해상도가 클수록 상당히 많은 메모리를 요구한다. 따라서, 이 방식은 내부 메모리의 용량이 적을 경우에는 내부 메모리에 모든 데이터를 저장할 수 없어서 외부 메모리를 이용한다. 이와 같이 외부 메모리를 이용할 경우에는 동작 속도가 빠른 프로세서에 비해 메모리의 접근 속도가 느려서 시스템 속도에 저하가 일어난다. 또한, 외부 메모리의 잦은 접근은 많은 전력 소비를 일으키므로, 전력 소비를 중요시하는 이동 단말 장치에는 상기 방식을 적용하기가 어려웠다.

- <16> 도 2b에 도시된 방식은 도 2a에서 설명된 방식의 단점을 보완하기 위해 이중 버퍼링(double buffering)을 이용한다. 이러한 이중 버퍼링을 효과적으로 사용하기 위해서는 메모리 간의 데이터 이동에 프로세서의 간섭이 없는 직접 메모리 접근(DMA : Direct Memory Access, 이하 'DMA'라 함) 방식이 적용되어야 한다. 즉, 하나의 매크로 블록이 수행될 때, 도 2b의 버퍼 1(buffer 1)에 저장한 참조 프레임을 사용하고, 이와 동시에 DMA를 이용하여 도 2b의 버퍼 2(buffer 2)에 프로세서의 간섭없이 다음 참조 프레임을 저장한다. 의도하는 매크로 블록에 대해 움직임 검출이 수행되면, 다음 매크로 블록에 대한 움직임 검출은 이미 DMA를 이용해서 이동된 다음 참조 프레임으로써 움직임 벡터를 찾는다.
- <17> 이동 단말 장치와 같은 임베디드 시스템에서는 프로세서의 성능이 이동형 배터리의 전력 관리 측면으로 인해 일반 프로세서와는 상당히 다른 주파수에서 동작한다. 또한, 내부 메모리도 상업적 측면으로 인해 그 용량이 작다. 따라서, 이와 같은 이동 단말 장치에서 복잡도가 큰 MPEG-4 비디오 부호기의 압축 알고리즘을 구현하기 위해서는 압축 시간 뿐만 아니라 전력을 많이 소비하는 움직임 검출 부분을 효과적으로 구현하는 것이 필요하다.
- <18> 기존에는 움직임 벡터를 추정할 때, 메모리의 종류나 크기에 관계없이 장치의 외부 메모리에 전체 이미지를 모두 저장하고, 두 프레임 간의 MSE 값 또는 SAD 값 등을 비교하여 최소값을 갖는 위치를 움직임 벡터로 하였다. 그러나, 적은 메모리 용량을 갖는 장치에서는 보다 효율적인 메모리 사용이 요구된다. 즉, 외부 메모리에 이미지를 저장하고 이를 사용하면 프로세서의 성능이 저하되고 전력 소비가 많이 유발되므로, 외부 메모리로의 접근을 최소화하고 외부 메모리의 장점인 블록 메모리 단위로 접근하여 메모리의 연속 접근을 행하여 내부 메모리를 보다 효과적으로 사용할 수 있는 방법이 요구된다.

【발명이 이루고자 하는 기술적 과제】

<19> 본 발명은 상기 설명한 종래의 기술적 과제를 해결하기 위한 것으로서, 메모리를 보다 효과적으로 사용하여 MPEG-4 비디오 부호기가 작동되도록 하는 이동 단말 장치에서의 움직임 검출 방법을 제공하는데 그 목적이 있다.

【발명의 구성 및 작용】

<20> 상기한 목적을 달성하기 위한 본 발명에 따른 이동 단말 장치에서의 움직임 검출 방법은, (a) 하나의 프레임에 대한 매크로 블록 단위의 가로 및 세로의 개수를 구하고, 매크로 블록 단위의 가로와 세로의 크기를 초기화하는 단계; (b) 상기 프레임의 세로 방향 소정 수의 데이터를 내부 메모리의 소정 수의 블록 버퍼로 각각 이동시키는 단계; (c) 상기 각 블록 버퍼에 저장된 데이터에 대해 세로 방향의 블록 버퍼의 위치를 업카운트 시켜가면서 움직임 검출을 수행하는 단계; (d) 세로 방향 블록 버퍼에 대한 움직임 검출이 완료되었는지 판단하고, 세로 방향 블록에 대한 움직임 검출이 완료될 경우, 매크로 블록 단위의 세로의 크기를 초기화시키고 가로 방향의 블록 버퍼의 위치를 업카운트시키는 단계; 및, (e) 가로 방향 블록 버퍼에 대한 움직임 검출이 완료되었는지 판단하고, 가로 방향 블록에 대한 움직임 검출이 완료되지 않았을 경우, 상기 단계 (b)로 점프하여 그 이후의 단계를 수행하는 단계를 포함하는 것을 특징으로 한다.

<21> 이하, 본 발명의 실시예에 따른 이동 단말 장치에서의 움직임 검출 방법을 첨부한 도면을 참조하여 상세히 설명하기로 한다.

<22> 도 3은 본 발명의 실시예에 따른 이동 단말 장치에서의 움직임 검출 방법이 적용되는 메모리 사용예를 도시한 도면으로서, 움직임 검출을 위한 연속적인 매크로 블록(13) X와 Y에 대해서 각각의 매크로 블록에 따른 참조 탐색 영역(11, 12)을 보여준다. 도3에 도시된 바와 같이, 현재 프레임의 연속하는 매크로 블록에 대해서 참조 탐색 영역에서는 중복된 메모리의 블록이 발생한다. 이 때, 기존의 방법에서는 참조 탐색 영역에 중복된 메모리 블록(X/Y)을 폐기하고, 다음 매크로 블록 움직임 검출시 폐기된 데이터를 갱신하게 된다. 그러나, 본 발명에서는 중복된 메모리 블록을 재사용하고 필요한 메모리 블록만 버퍼에 저장하여 메모리의 접근을 줄인다.

<23> 또한, 도 3에 도시된 바와 같이, 본 발명의 실시예에 따른 이동 단말 장치에서의 움직임 검출 방법에서는, 버퍼의 크기를 줄이는 대신에 버퍼의 개수를 늘려서 기존의 이중 버퍼링 방식에서 사용되는 내부 메모리의 크기를 줄여 작은 내부 메모리를 보다 효과적으로 사용하는 것이 가능하다. 즉, 도 4a에 도시된 기존의 이중 버퍼를 이용한 방식에서는 내부 메모리의 참조 탐색 영역을 위해 하나의 매크로 블록에 대한 움직임 검출 시에 9개의 매크로 블록이 사용되고 폐기되지만, 도 4b에서와 같이, 본 발명에서 제안된 방식에서는 3개의 매크로 블록으로 구성된 P[3]이라는 하나의 버퍼가 더 구비되어 순환 버퍼형식을 이용하며, 메모리 간의 데이터 이동은 3개의 매크로 블록만을 메모리 이동시켜 움직임 검출이 가능하다.

<24> A, B, C, D에 대한 매크로 블록에 대한 움직임 검출이 수행될 때, 데이터의 이동크기는 도 4b에 제시된 바와 같다. 기존의 방식에서 이중 버퍼를 사용하는 경우에는 버퍼 1(buffer 1) 및 버퍼 2(buffer 2)를 각각 두 번씩 사용하므로 9개의 매크로 블록에 4를 곱한 36개의 매크로 블록의 데이터 이동이 발생한다. 그러나, 본 발명에서 제안된 순환버퍼의 경우에는 18개의 매크로 블록의 데이터 이동이 있어, 기존의 방식에 비해 데이터 이동량을 절반으로 줄이고

도 움직임 검출을 수행할 수 있다. 매크로 블록의 개수가 커지면 이 둘의 데이터 이동비는 더 작아져서 1/3의 값으로 수렴한다. 즉, 간단히 매크로 블록 개수를 N 이라 하면, 이중 버퍼의 경우에는 $N*9$ 이 되고, 순환버퍼의 경우에는 $(N+2)*3$ 이 된다.

<25> 도 3에 도시된 움직임 검출 방법을 프레임 측면에서 고려하면, 도 5의 (a)에 도시된 바와 같이, 프레임의 매크로 블록의 움직임을 세로방향(column)으로 먼저 진행하고 다시 가로방향으로 초기화하여 진행하는 방식과, 도 5의 (b)에 도시된 바와 같이, 프레임의 매크로 블록의 움직임을 가로방향(row)으로 먼저 진행하고 다시 세로방향으로 초기화하여 진행하는 방식이 가능하다. 도 5의 (c)는 먼저 세로방향으로 움직임 검출을 수행할 때, 버퍼의 할당과 데이터의 이동을 보여주는 도면이며, 도 5의 (d)는 먼저 가로방향으로 움직임 검출을 수행할 때, 버퍼의 할당과 데이터의 이동을 보여주는 도면이다. 도 5의 (c)와 (d)에 도시된 "t"의 값은 움직임 검출을 원하는 매크로 블록의 위치를 나타낸다.

<26> 다음으로, 도 6의 순서도를 참조하여 본 발명의 실시예에 따른 이동 단말 장치에서의 움직임 검출 방법을 설명한다.

<27> 먼저, 도 5의 (a)와 같이, 세로 방향에 대한 움직임 검출은 도 6의 순서도에 따라 진행된다. 즉, 하나의 프레임에 대해 움직임 검출 알고리즘이 시작되면(S110), 한 프레임에 대한 매크로 블록 단위의 가로 및 세로의 개수를 프레임의 가로(Width)와 세로(Height)의 픽셀 개수로부터 구하고, 매크로 블록 단위의 가로와 세로의 초기값을 '0'으로 설정한다(S111). 프레임의 제일 처음 시작점 혹은 세로 방향의 시작점에서는 우선 3개의 매크로 블록으로 묶인 버퍼 3개($P[0], P[1], P[2]$)를 외부 메모리로부터 데이터를 가져와 내부 메모리에 저장한다(S112). 위의 3개의 버퍼는 현재 프레임의 첫 매크로 블록의 참조 탐색 영역이 되며, 이에 따라 그 매크로 블록에 대한 움직임 검출이 수행된다(S114). 원하는 하나의 매크로 블록에 대해 움직임

검출 수행이 끝나면, 세로 방향의 다음 매크로 블록에 대한 움직임 검출을 수행하기 위해 세로 방향의 매크로 블록 수를 하나 업카운트시킨다(S115). 그 다음에, 매크로 블록이 현재 프레임의 세로방향에 대한 마지막 매크로 블록인지 확인한다(S116). 상기 단계(S116)에서 매크로 블록이 현재 프레임의 세로방향에 대한 마지막 매크로 블록이 아니면, 다음 세로방향의 매크로 블록의 움직임 검출을 수행하기 위해 $P[(\text{세로방향의 매크로 블록 수}+2)\%4]$ 위치의 버퍼에 참조 탐색 영역을 위한 3개의 매크로 블록 데이터를 이동시킨다. 상기 단계(S116)에서 하나의 세로 방향 블록에 대한 움직임 검출이 끝난 것으로 판단되면, 세로방향에 대한 값(Column)을 초기화시키고 가로방향에 대한 값(Row)을 하나 업카운트시킨다(S117). 이제, 가로방향의 블록이 종료되었는지를 판단하여 가로방향의 움직임 검출이 완료되었는지를 확인한다(S118). 상기 단계(S118)에서 가로방향의 움직임 검출이 완료되지 않았으면, 상기 단계(S112)로 점프하여 움직임 검출 알고리즘을 계속 수행하며, 상기 단계(S118)에서 가로방향의 움직임 검출이 완료되었으면, 움직임 검출 알고리즘이 종료된다(S119).

<28> 도 5의 (b)에서와 같이 가로방향의 블록에 대해 먼저 움직임 검출이 수행될 경우에는 상기 도 6에 도시된 순서도에서 세로 방향과 가로방향에 대한 내용을 서로 바꾸면 동일한 방법으로 구현되므로 자세한 설명은 생략한다.

<29> 버퍼에 저장될 데이터는 3차원 배열형식으로 저장되는데, 배열 형식에 따라 SAD값을 구하는 방법이 다르다. 세로방향의 경우에는 도 7a에 도시된 바와 같이, 3개의 매크로 블록이 $48*16$ 의 배열을 만들어 하나의 버퍼를 형성하고 각 움직임 검출이 일어날 때마다 버퍼 하나의 데이터 이동으로 하나의 움직임 벡터를 구할 수 있도록 한다. 가로방향의 경우에는 $16*48$ 의 배열을 구성해서 하나의 버퍼를 형성하여 총 4개의 버퍼를 순환 형식으로 설정하며, 역시 하나의 움직임 검출마다 하나의 버퍼 크기의 데이터 이동이 일어나도록 한다.

<30> 도 8a 및 도 8b는 탐색 구간을 $[-16, 15]$ 로 설정했을 때의 원하는 매크로 블록(P_c)에 대한 참조 탐색 영역(11)($P_{ref}[x-1], P_{ref}[x], P_{ref}[x+1]$)을 나타내고 있다. 두 매크로 블록의 유사도 SAD를 구하려면 두 개의 버퍼에 걸쳐있는 경우가 있으므로 이에 대한 고려가 있어야 한다. 즉, 도 8a에 도시된 바와 같이, 움직임 좌표 (i, j) 에 대해서 유사도 SAD는 13a의 SAD_1 과 13b의 SAD_2 의 합으로 구해지며, 움직임 좌표 (i, j) 에 따른 SAD_1 과 SAD_2 의 값은 아래의 수학적식 1과 같다.

<31>

$0 \leq j < 16$ 원하는 매크로 블록이 $P_{ref}[(x-1)\%4]$ 과 $P_{ref}[x\%4]$ 사이에 존재할 때

$$\begin{cases} SAD_1 = \sum_{k=0}^{15-j} \sum_{l=0}^{15} |P_c[k][l] - P_{ref}[(x-1)\%4][k+j][l]| & \dots\dots\dots 13(a) \\ SAD_2 = \sum_{k=15-j}^{15} \sum_{l=0}^{15} |P_c[k][l] - P_{ref}[x\%4][k-(15-j)][l]| & \dots\dots\dots 13(b) \end{cases}$$

$16 \leq j < 32$ 원하는 매크로 블록이 $P_{ref}[x\%4]$ 과 $P_{ref}[(x+1)\%4]$ 사이에 존재할 때

$$\begin{cases} SAD_1 = \sum_{k=0}^{15-j} \sum_{l=0}^{15} |P_c[k][l] - P_{ref}[x\%4][k+j][l]| & \dots\dots\dots 13(a) \\ SAD_2 = \sum_{k=15-j}^{15} \sum_{l=0}^{15} |P_c[k][l] - P_{ref}[(x+1)\%4][k-(15-j)][l]| & \dots\dots\dots 13(b) \end{cases}$$

【수학적식 1】

<32> 상기 수학적식 1에서 i 는 하나의 버퍼에 속하므로 i 의 증가에 따른 식의 변화가 없다. 그러나, j 의 경우에는 다른 버퍼 영역에 속할 수 있으므로 상기 수학적식 1과 같이 두 가지 범위를 두어야 한다. 상기 매크로 블록(P_c)에 대한 움직임 검출에 의해 움직임 벡터 값이 구해지면 $P[(x+2)\%4]$ 의 버퍼에 다음 매크로 블록에 대한 움직임 검출을 위해 하나의 버퍼 데이터가 이동되어 이전의 중복된 버퍼와 하나가 되어 참조 탐색 영역(6)이 형성된다.

<33> 도 8b에서의 가로방향 움직임 검출에서도 탐색구간에 대한 SAD 값을 구하는 방식은 도 8a에서와 유사한 방식이다. 이 때의 SAD 값을 구하는 실 예는 아래의 수학적식 2와 같다. 가로

방향은 j 값이 하나의 버퍼에 존재하므로 j 에 따른 식의 변화가 없고 i 값에 따라 SAD값을 구하는 범위가 결정된다.

<34>

$0 \leq i < 16$ 원하는 매크로 블록이 $P_{ref}[(x-1)\%4]$ 과 $P_{ref}[x\%4]$ 사이에 존재할 때

$$\begin{cases} SAD_1 = \sum_{k=0}^{15} \sum_{l=0}^{15-i} |P_c[k][l] - P_{ref}[(x-1)\%4][k][l+i]| & \dots\dots\dots 14(a) \\ SAD_2 = \sum_{k=0}^{15} \sum_{l=15-i}^{15} |P_c[k][l] - P_{ref}[x\%4][k][l-(15-i)]| & \dots\dots\dots 14(b) \end{cases}$$

$16 \leq i < 32$ 원하는 매크로 블록이 $P_{ref}[x\%4]$ 과 $P_{ref}[(x+1)\%4]$ 사이에 존재할 때

$$\begin{cases} SAD_1 = \sum_{k=0}^{15} \sum_{l=0}^{15-i} |P_c[k][l] - P_{ref}[x\%4][k][l+i]| & \dots\dots\dots 14(a) \\ SAD_2 = \sum_{k=15}^{15} \sum_{l=15-i}^{15} |P_c[k][l] - P_{ref}[(x+1)\%4][k][l-(15-i)]| & \dots\dots\dots 14(b) \end{cases}$$

【수학식 2】

<35>

위 예에서는 탐색구간을 $[-16,15]$ 에 대해서 설명했는데, 움직임이 적은 영상의 경우 탐색구간이 이보다 작은 $[-8,7]$ 로도 가능하다. 이러한 탐색 구간도 많이 사용되며, 이에 대한 세로방향과 가로방향에 대한 참조영상과 버퍼 데이터의 이동은 도 9에 도시되어 있다. 상기도 9에 도시된 바와 같이, 참조 탐색 영역이 작으므로 버퍼의 할당도 작아지고 순환버퍼의 크기도 작아진다.

<36>

탐색구간의 변화에 따라 내부 메모리의 버퍼 할당이 달라지는데, 탐색구간의 변화에도 버퍼의 할당을 일정하게 유지하려면 도 10에 도시된 바와 같이 움직임 검출을 수행하면 된다. 즉, 버퍼의 이동이 있을 때 하나의 움직임 검출을 하는 것이 아니라, 두개의 움직임 검출(X,Y)을 바로 수행하면 된다. 세로방향과 가로방향에 대한 움직임 검출은 역시 위에서 설명한 것과 동일하다.

【발명의 효과】

- <37> 이상 설명된 바와 같이, 본 발명에 따른 움직임 검출 방법이 구현되면, 적은 내부 메모리를 가진 이동 단말 장치에서도 효과적으로 메모리를 접근하여 속도 성능이 좋아진다. 뿐만 아니라, 전력을 많이 소비하는 외부 메모리의 접근을 낮추고 적은 전력을 소비하는 칩 내부 메모리를 사용하므로 이동형 배터리의 사용 가능한 상태를 장기간 동안 유지할 수 있으므로 이동 단말 장치에 보다 향상된 휴대 편리성을 제공한다.
- <38> 이상으로 설명한 것은 본 발명에 따른 이동 단말 장치에서의 움직임 검출 방법을 실시하기 위한 하나의 실시예에 불과한 것으로서, 본 발명은 상기한 실시예에 한정되지 않고, 이하의 특허청구의 범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변경 실시가 가능한 범위까지 본 발명의 기술적 정신이 미친다고 할 것이다.

【특허청구범위】**【청구항 1】**

(a) 하나의 프레임에 대한 매크로 블록 단위의 가로 및 세로의 개수를 구하고, 매크로 블록 단위의 가로와 세로의 크기를 초기화하는 단계;

(b) 상기 프레임의 세로 방향 소정 수의 데이터를 내부 메모리의 소정 수의 블록 버퍼로 각각 이동시키는 단계;

(c) 상기 각 블록 버퍼에 저장된 데이터에 대해 움직임 검출을 수행하고, 세로 방향의 매크로 블록 수를 업카운트시키는 단계;

(d) 세로 방향 블록 버퍼에 대한 움직임 검출이 완료되었는지 판단하고, 세로 방향 블록에 대한 움직임 검출이 완료될 경우, 매크로 블록 단위의 세로의 크기를 초기화시키고 가로 방향의 블록 버퍼의 위치를 업카운트시키는 단계; 및

(e) 가로 방향 블록 버퍼에 대한 움직임 검출이 완료되었는지 판단하고, 가로 방향 블록에 대한 움직임 검출이 완료되지 않았을 경우, 상기 단계 (b)로 점프하여 그 이후의 단계를 수행하는 단계를 포함하는

이동 단말 장치에서의 움직임 검출 방법.

【청구항 2】

제1항에 있어서,

상기 단계 (d)에서 세로 방향의 움직임 검출이 완료되지 않은 경우에는 %연산에 의해 얻어지는 블록 위치에 소정 단위의 매크로 블록 데이터를 이동시키는

이동 단말 장치에서의 움직임 검출 방법.

【청구항 3】

제2항에 있어서,

상기 %연산은 매크로 블록의 단위가 3개 일 경우, ((세로방향의 매크로 블록 수 + 2) % 4)의 수식에 의해 블록 위치가 결정되는

이동 단말 장치에서의 움직임 검출 방법.

【청구항 4】

제1항에 있어서,

탐색 구간이 $[-16, 15]$ 일 경우, 세로 방향을 48×16 의 배열로 형성하고 가로 방향을 16×48 의 배열로 정렬하여 데이터의 이동시 블록화하고, 각각의 버퍼가 순환버퍼가 되도록 %연산을 수행하여 나머지 값으로 버퍼의 주소를 구하며, SAD 값을 구하는데 있어 순환 버퍼 상에 (i, j) 에 따른 유사도를 구하는

이동 단말 장치에서의 움직임 검출 방법.

【청구항 5】

제1항에 있어서,

탐색 구간이 $[-8, 7]$ 일 경우, 세로 방향에 대해서는 32×16 의 배열로 형성하고 가로방향에 대해서는 16×32 의 배열로 정렬하여 데이터 이동시 블록화하고, 각각

의 버퍼가 순환버퍼가 되도록 %연산을 수행하여 나머지 값으로 버퍼의 주소를 구하며, SAD 값을 구하는데 있어 순환 버퍼 상에 (i, j) 에 따른 유사도를 구하는

이동 단말 장치에서의 움직임 검출 방법.

【청구항 6】

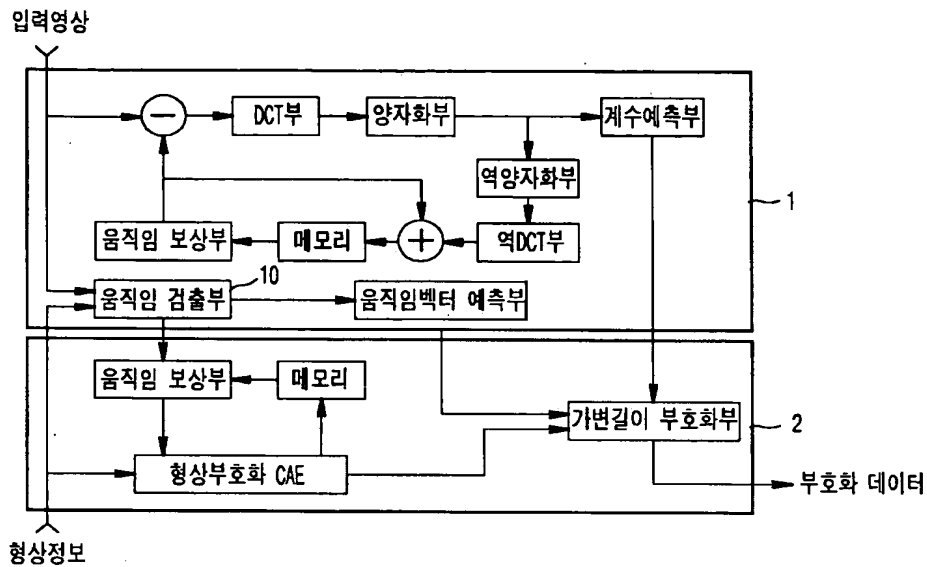
제1항에 있어서,

탐색 구간이 $[-16, 15]$ 에 대한 고정된 순환 버퍼의 크기와 할당에 있어서 탐색구간이 $[-8, 7]$ 일 때 움직임 검출을 수행하는

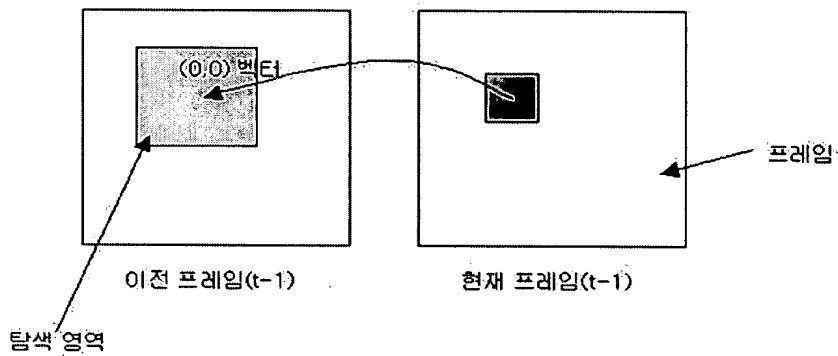
이동 단말 장치에서의 움직임 검출 방법.

【도면】

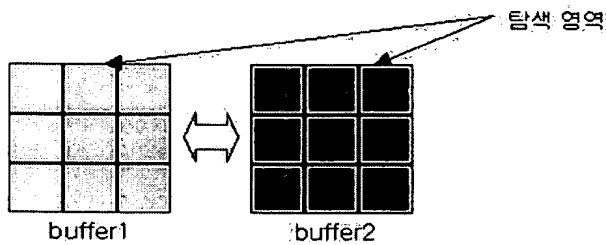
【도 1】



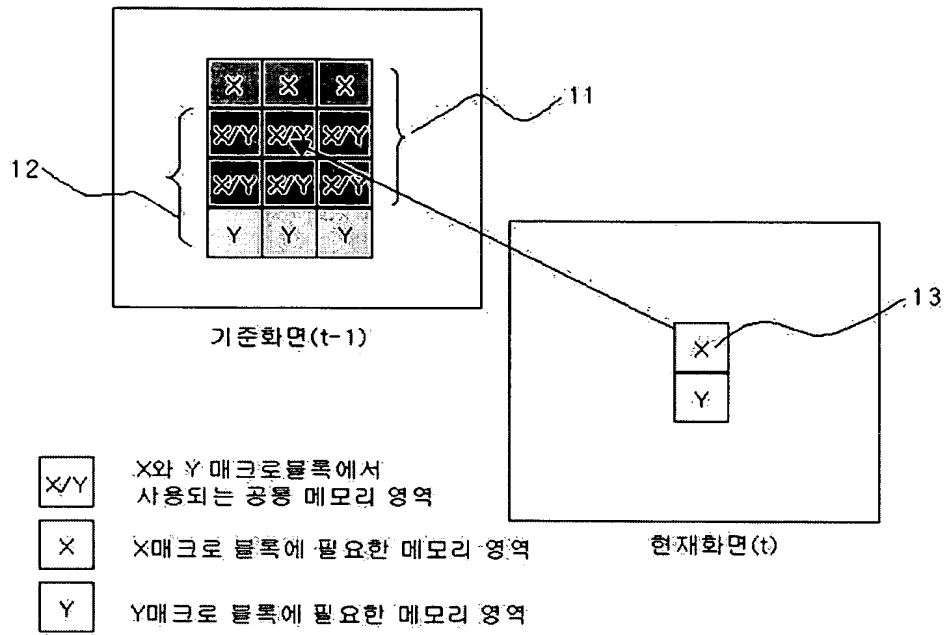
【도 2a】



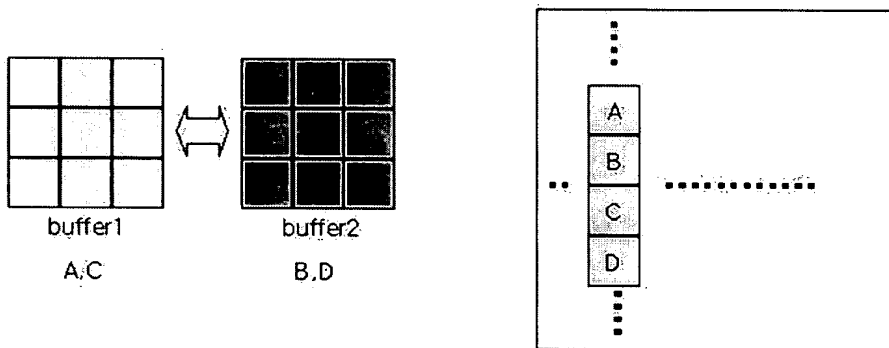
【도 2b】



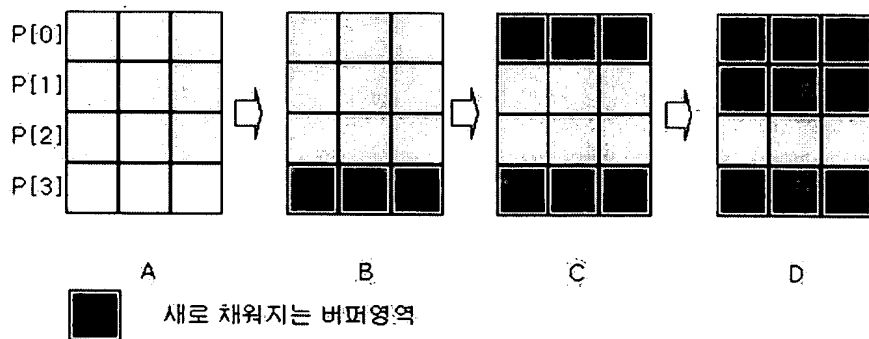
【도 3】



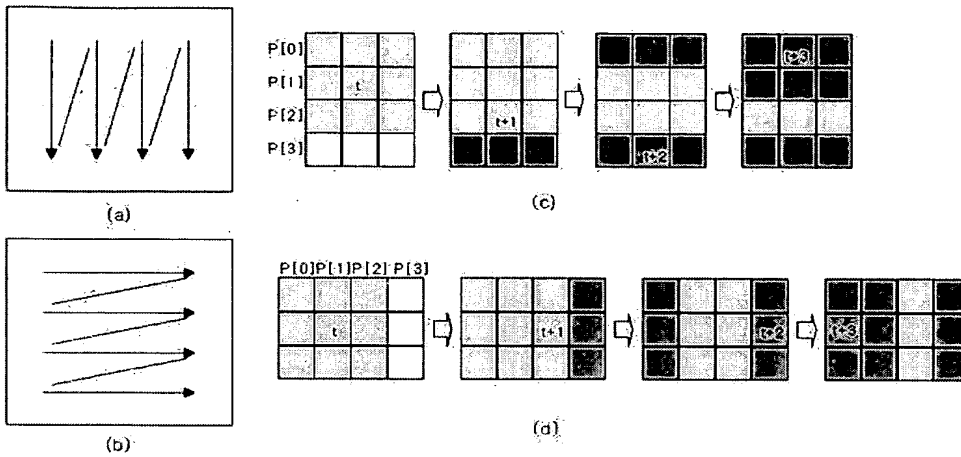
【도 4a】



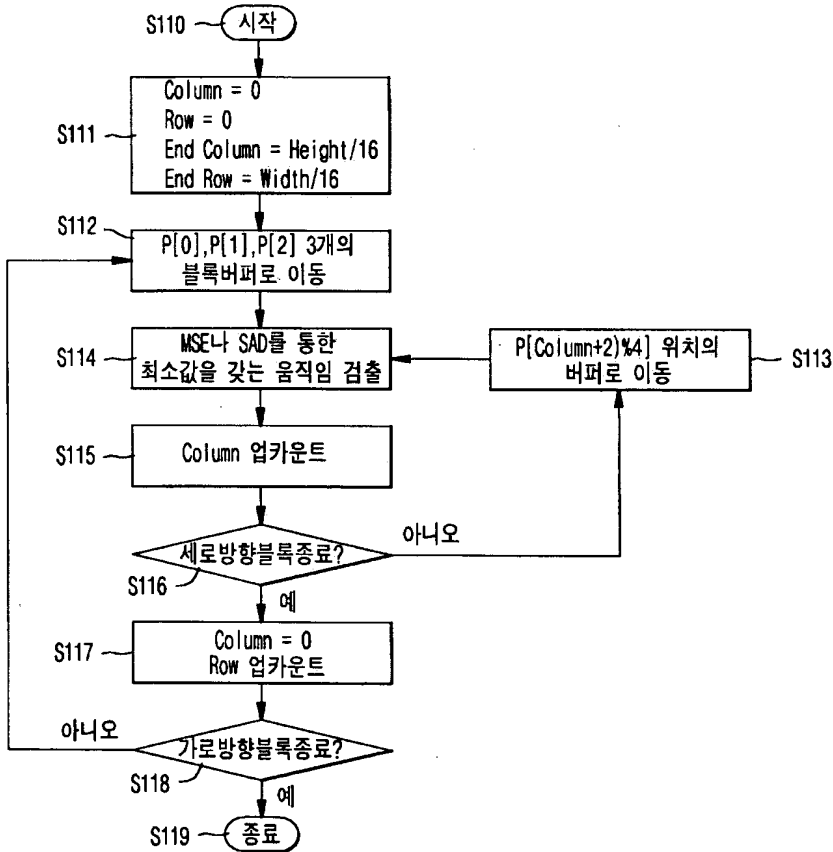
【도 4b】



【도 5】



【도 6】



【도 7a】

 $P[x]$

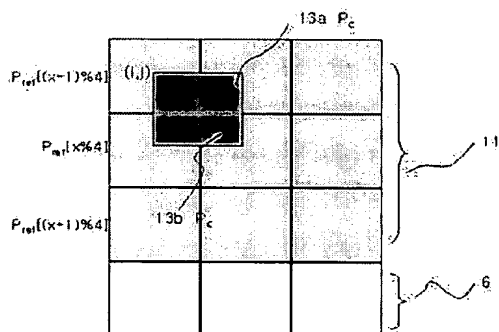
[0]	[0], [1], [2], ... [46], [47]
[1]	
[2]	
...	
[15]	

【도 7b】

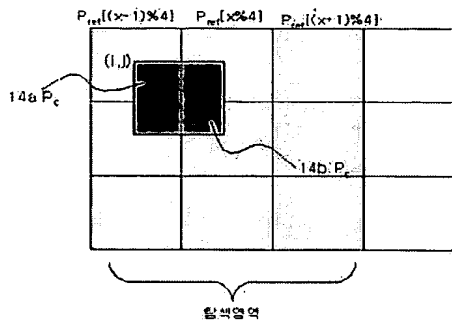
 $P[x]$

[0]	[0], [1], [2], ..., [15]
[1]	
[2]	
...	
[47]	

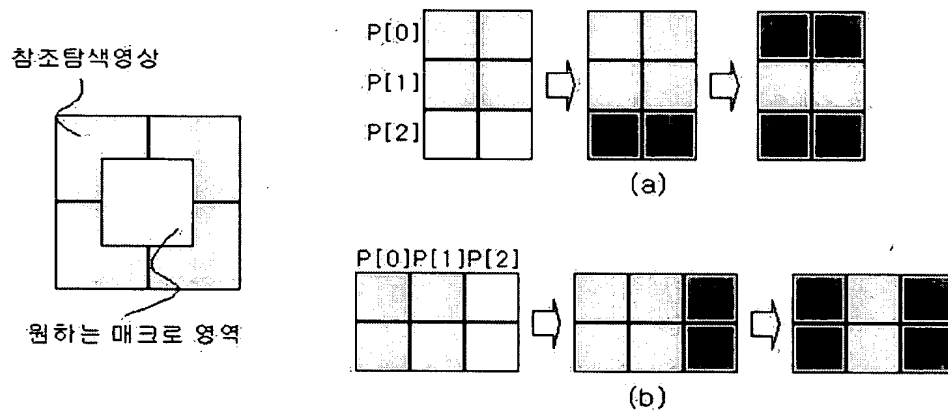
【도 8a】



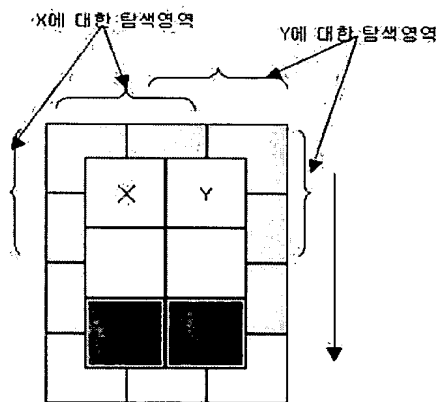
【도 8b】



【도 9】



【도 10a】



【도 10b】

